We’re Hiring!
Common Terms
A failure occurs when a system or component of a system that is not behaving within normal operating parameters.
Incident

An incident is a failure that results in a visible impact to a user of the system.
MTTR

Mean Time To Recovery.
We’re Hiring!
Rules
Rules

• Optimize for MTTR
• Alerts must be *actionable*.
• Ruthlessly eliminate false positives.
• Don’t make me think.
• Aviate, Navigate, Communicate.
• *Beware incident fatigue*.
• Did I mention you should communicate?
Heroku On-Call
Heroku On-Call

- Every engineer participates.
- Three Roles: Primary, Backup, Incident Commander.
- Positive acknowledgment via alert.
- On-call hand-off email.
- Scheduling is hard.
Heroku On-Call: Primary

- Respond to alerts within 5 minutes.
- Resolve routine alerts via operational playbooks.
- Escalate to the Incident Commander when:
  - Incident affects more than 1,000 apps.
  - 10 minutes without resolution in sight.
  - Next steps unclear.
- On-call handoff email.
Heroku On-Call: Backup

- Respond to alerts within 10 minutes.
- See Primary.
Heroku On-Call: Incident Commander

• Evaluate issue and resolve or call for help.
• Coordinate actions across responders.
• Notify support and product teams.
• Manage external status updates.
• Conduct and communication post-mortem reviews.
We’re Hiring!
Playbooks
Playbooks

• Linked directly from the failure notification.
• Describe the alert.
• How can I verify that the failure is occurring?
• Actions:
  • How to resolve?
  • When to escalate?
• How can I verify that the failure is resolved?
• What now? Customer notification, team email, etc.
Runtime Failure

- Linked directly from the failure notification.
- How to identify.
- Steps to report a false positive.
- Steps to destroy a runtime.
- Escalation.
- Notes.
Incidents
Runtime Failure

• Nagios detects that the Runtime instance is not responding and performs a couple of retries.

• Primary on-call engineer is paged with a playbook link and acknowledges the alert.

• Runtime instance appears to be unreachable, so the instance is terminated through our admin GUI.

• The incident is marked resolved and an email is sent to our runtime team letting them know.

• Zzzzzzzzz
Shen Failure

- Nagios detects a problem with a shared database server and notifies the Primary on-call engineer.

- Primary on-call engineer is paged with a playbook link and acknowledges the alert.

- Engineer determines instance is down and opens a public status event with a pre-defined message.

- Instance is recovered.

- Status event is closed and the incident resolved.

- Zzzzzzzzz
Skynet

• 67 hours of total status event time.
• Shift work was a must, but we can do better.
• Largely at the mercy of Amazon for recovery, but still needed to communicate.
• All hands on deck, we got a lot of help from the other engineering teams and support team.
• Were able to recover some customers quickly, but we didn’t have a great way to define who was impacted.
• ... and I was on vacation!
Next?
Next?

• Much better metrics.
• DSL/Annotations for runnable playbooks?
• Auto-healing for common failures.
• Consolidated “Ops Console”.
• Remove Primary on-call and escalate immediately?
• Fine grained status events?
Bonus Round
Thanks!
(Did I mention we’re hiring?)